

[illegible][illegible]

```

BBBBBBBBB  SSSSSSSS  DDDDDDDD  RRRRRRRR  IIIIII  VV  VV  EEEEEEEEE  RRRRRRRR
BBBBBBBBB  SSSSSSSS  DDDDDDDD  RRRRRRRR  IIIIII  VV  VV  EEEEEEEEE  RRRRRRRR
BB      BB  SS      DD      DD  RR      RR  II      VV  VV  EE      RR      RR
BB      BB  SS      DD      DD  RR      RR  II      VV  VV  EE      RR      RR
BB      BB  SS      DD      DD  RR      RR  II      VV  VV  EE      RR      RR
BB      BB  SS      DD      DD  RR      RR  II      VV  VV  EE      RR      RR
BBBBBBBBB  SSSSSS  DD      DD  RRRRRRRR  II      VV  VV  EEEEEEEE  RRRRRRRR
BBBBBBBBB  SSSSSS  DD      DD  RRRRRRRR  II      VV  VV  EEEEEEEE  RRRRRRRR
BB      BB  SS      DD      DD  RR      RR  II      VV  VV  EE      RR      RR
BB      BB  SS      DD      DD  RR      RR  II      VV  VV  EE      RR      RR
BB      BB  SS      DD      DD  RR      RR  II      VV  VV  EE      RR      RR
BBBBBBBBB  SSSSSSSS  DDDDDDDD  RR      RR  IIIIII  VV  VV  EEEEEEEEE  RR      RR
BBBBBBBBB  SSSSSSSS  DDDDDDDD  RR      RR  IIIIII  VV  VV  EEEEEEEEE  RR      RR

LL  IIIIII  SSSSSSSS
LL  IIIIII  SSSSSSSS
LL  II      SS
LL  II      SS
LL  II      SS
LL  II      SS
LL  II      SSSSSS
LL  II      SSSSSS
LL  II      SS
LL  II      SS
LL  II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS

```

E 13
16-Sep-1984 00:00:33
5-Sep-1984 13:48:56

VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]BSDRIVER.FOR;1

Page 1

```
0001 C
0002 C Version: 'V04-000'
0003 C
0004 C*****
0005 C*
0006 C* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0007 C* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0008 C* ALL RIGHTS RESERVED.
0009 C*
0010 C* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0011 C* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0012 C* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0013 C* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0014 C* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0015 C* TRANSFERRED.
0016 C*
0017 C* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0018 C* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0019 C* CORPORATION.
0020 C*
0021 C* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0022 C* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0023 C*
0024 C*
0025 C*****
0026 C
0027 C
0028 c Author Brian Porter Creation date: 16-JUL-1982
0029 c
0030 c++
0031 c Functional description:
0032 c
0033 c This module displays events logged by the BSDRIVER (dt07).
0034 c
0035 c Modified by:
0036 c
0037 c V03-003 SAR0216 Sharon A. Reynolds, 28-Mar-1984
0038 c Changed the call to UCB$L_OWNUIC to ORB$L_OWNER.
0039 c
0040 c V03-002 SAR0065 Sharon A. Reynolds, 20-Jun-1983
0041 c Changed the carriage control in the 'format' statements
0042 c for use with ERF.
0043 c
0044 c V03-001 SAR0037 Sharon A. Reynolds, 8-Jun-1983
0045 c Removed brief/cryptic support.
0046 c
0047 c**
0048 c--
0049 c
0050 Subroutine BSDRIVER (lun)
0051
0052
0053 include 'src$:msghdr.for /nolist'
0112 include 'src$:deverr.for /nolist'
0213
0214
0215 byte lun
```

```
0216
0217 integer*4      ucb$b_bs_errmsg
0218 integer*4      ucb$l_devdepend
0219 integer*4      ucb$l_bs_cur
0220 integer*4      ucb$l_bs_pre
0221
0222 equivalence    (emb$l_dv_reg sav(0),ucb$b_bs_errmsg)
0223 equivalence    (emb$l_dv_reg sav(1),ucb$l_devdepend)
0224 equivalence    (emb$l_dv_reg sav(2),ucb$l_bs_cur)
0225 equivalence    (emb$l_dv_reg sav(3),ucb$l_bs_pre)
0226
0227
0228 character*32    vlucb$l_devdepend(0:12)
0229 data vlucb$l_devdepend(0) /'"OWNER", CURRENT PROCESS*'/
0230 data vlucb$l_devdepend(1) /'"ATTENTION AST ENABLED*'/
0231 data vlucb$l_devdepend(2) /'"SWITCHED BUS IN USE*'/
0232 data vlucb$l_devdepend(3) /'"PORT HAS PRIMARY STATUS*'/
0233 data vlucb$l_devdepend(4) /'"CURRENTLY IN PROGRAM MODE*'/
0234 data vlucb$l_devdepend(5) /'"CURRENTLY IN MANUAL MODE*'/
0235 data vlucb$l_devdepend(6) /'"DRIVER STATUS INITIALIZED*'/
0236 data vlucb$l_devdepend(7) /'"SWITCHED DEVICES MARKED OFFLINE*'/
0237 data vlucb$l_devdepend(8) /'"SWITCHED DEVICES MARKED ON-LINE*'/
0238 data vlucb$l_devdepend(9) /'"SWITCHED BUS DISCONNECT-IN-PROG*'/
0239 data vlucb$l_devdepend(10) /'"SWITCHED BUS CONNECTED*'/
0240 data vlucb$l_devdepend(11) /'"UBA" INITIALIZE-IN-PROGRESS*'/
0241 data vlucb$l_devdepend(12) /'"DEVICE INTERRUPT DISABLED*'/
0242
0243
0244 character*31    vlcsr(0:15)
0245 data vlcsr(0)    /'"REQUEST*'/
0246 data vlcsr(1)    /'"HOLD*'/
0247 data vlcsr(2)    /'"REQUEST LINE #0*'/
0248 data vlcsr(3)    /'"REQUEST LINE #1*'/
0249 data vlcsr(4)    /'"REQUEST LINE #2*'/
0250 data vlcsr(5)    /'"REQUEST LINE #3*'/
0251 data vlcsr(6)    /'"INTERRUPT ENABLE*'/
0252 data vlcsr(7)    /'"PORT CONNECTED TO SWITCHED BUS*'/
0253 data vlcsr(8)    /'"PORT REQUESTING MASTERSHIP*'/
0254 data vlcsr(9)    /'"GENERATE RESET PULSE*'/
0255 data vlcsr(10)   /'"PORT IN MANUAL MODE*'/
0256 data vlcsr(11)   /'"POWER-OK OTHER PORTS*'/
0257 data vlcsr(12)   /'"EXTERNAL INTERRUPT*'/
0258 data vlcsr(13)   /'"SWITCHED BUS ACTIVE*'/
0259 data vlcsr(14)   /'"SWITCHED BUS POWER-FAILURE*'/
0260 data vlcsr(15)   /'"TIMEOUT*'/
0261
0262
0263 call frctof (lun)
0264
0265 call dhead1 (lun,'UBA DT07')
0266
0267 call linchk (lun,1)
0268
0269 call ucb$b_bs_errmsg (lun,ucb$b_bs_errmsg)
0270
0271 call linchk (lun,2)
0272
```

```
0273      write(lun,20) 'DT07 "CSR", CURRENT CONTENTS'
0274      format(/' ',a)
0275      call linchk (lun,2)
0276
0277      write(lun,25) ucb$l_bs_cur
0278      format(/' ',t8,'UCB$L_BS_CUR',t24,z8.8)
0279      call output (lun,ucb$l_bs_cur,vlcsr,0,0,15,'0')
0280
0281      call linchk (lun,2)
0282
0283      write(lun,20) 'DT07 "CSR", PREVIOUS CONTENTS'
0284
0285      call linchk (lun,2)
0286
0287      write(lun,30) ucb$l_bs_pre
0288      format(/' ',t8,'UCB$L_BS_PRE',t24,z8.8)
0289      call output (lun,ucb$l_bs_pre,vlcsr,0,0,15,'0')
0290
0291      call linchk (lun,1)
0292
0293      write(lun,32)
0294      format(' ',i)
0295
0296      call orb$l_owner (lun,emb$l_dv_ownuic)
0297
0298      call ucb$l_char (lun,emb$l_dv_char)
0299
0300      call ucb$w_sts (lun,emb$w_dv_sts)
0301
0302      call linchk (lun,1)
0303
0304      write(lun,35) ucb$l_devdepend
0305      format(' ',t8,'UCB$L_DEVDEPEND',t24,z8.8)
0306      call output (lun,ucb$l_devdepend,vlucb$l_devdepend,0,0,12,'0')
0307
0308      call ucb$l_opcnt (lun,emb$l_dv_opcnt)
0309
0310      call ucb$w_errcnt (lun,emb$w_dv_errcnt)
0311
0312      if (emb$w_hd_entry .ne. 98) then
0313
0314      call linchk (lun,1)
0315
0316      write(lun,32)
0317
0318      if (emb$w_dv_func .eq. 2) then
0319
0320      call irp$w_func (lun,emb$w_dv_func,'IO$_READEXT*')
0321
0322      else if (emb$w_dv_func .eq. 5) then
0323
0324      call irp$w_func (lun,emb$w_dv_func,'IO$_DISCONNECT*')
0325
0326
0327
0328
0329
```

```
0330      else if (emb$w_dv_func .eq. 50) then
0331
0332      call irp$w_func (lun,emb$w_dv_func,'IOS_CONNECT*')
0333      else
0334
0335      call irp$w_func (lun,emb$w_dv_func,'QIO FUNCTION CODE*')
0336      endif
0337
0338      call irp$l_pid (lun,emb$l_dv_rapid)
0339
0340      call irp$q_iosb (lun,emb$l_dv_iosb1)
0341      endif
0342
0343      return
0344
0345      end
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	625	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	240	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	1340	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3 EMB	512	PIC OVR REL GBL SHR NOEXE RD WRT LONG
Total Space Allocated	2717	

ENTRY POINTS

Address	Type	Name
0-00000000		BSDRIVER

VARIABLES

Address	Type	Name	Address	Type	Name
3-0000001C	L*1	EMB\$B_DV_CLASS	3-00000010	L*1	EMB\$B_DV_ERTCNT
3-00000011	L*1	EMB\$B_DV_ERTMAX	3-0000003E	L*1	EMB\$B_DV_NAMLANG
3-0000003A	L*1	EMB\$B_DV_SLAVE	3-0000001D	L*1	EMB\$B_DV_TYPE
3-00000036	I*4	EMB\$L_DV_CHAR	3-00000012	I*4	EMB\$L_DV_IOSB1
3-00000016	I*4	EMB\$L_DV_IOSB2	3-00000026	I*4	EMB\$L_DV_MEDIA
3-0000004E	I*4	EMB\$L_DV_NUMREG	3-0000002E	I*4	EMB\$L_DV_OPCNT
3-00000032	I*4	EMB\$L_DV_OWNUIC	3-0000001E	I*4	EMB\$L_DV_RQPID
3-00000000	I*4	EMB\$L_HD_SID	3-0000003F	CHAR	EMB\$T_DV_NAME
3-00000024	I*2	EMB\$W_DV_BCNT	3-00000022	I*2	EMB\$W_DV_BOFF
3-0000002C	I*2	EMB\$W_DV_ERRCNT	3-0000003C	I*2	EMB\$W_DV_FUNC
3-0000001A	I*2	EMB\$W_DV_STS	3-0000002A	I*2	EMB\$W_DV_UNIT
3-00000004	I*2	EMB\$W_HD_ENTRY	3-0000000E	I*2	EMB\$W_HD_ERRSEQ
AP-00000004a	L*1	LUN	3-00000052	I*4	UCB\$B_BS_ERRMSG
3-0000005A	I*4	UCB\$L_BS_CUR	3-0000005E	I*4	UCB\$L_BS_PRE

3-00000056 1*4 UCB\$\$_DEVDEPEND

ARRAYS

Address	Type	Name	Bytes	Dimensions
3-00000000	L*1	EMB	512	(0:511)
3-00000052	1*4	EMB\$_DV_REGS	420	(0:104)
3-00000006	1*4	EMB\$_HD_TIME	8	(2)
2-000001A0	CHAR	V1CSR	496	(0:15)
2-00000000	CHAR	V1UCB\$_DEVDEPEND	416	(0:12)

LABELS

Address	Label	Address	Label	Address	Label	Address	Label	Address	Label
1-00000095	20'	1-0000009B	25'	1-000000B5	30'	1-000000CF	32'	1-000000D4	35'

FUNCTIONS AND SUBROUTINES REFERENCED

Type Name

DHEAD1
IRP\$_IOSB
ORB\$_OWNER
UCB\$_CHAR
UCB\$_STS

Type Name

FRCTOF
IRP\$_FUNC
OUTPUT
UCB\$_OPCNT

Type Name

IRP\$_PID
LINCHR
UCB\$_BS_ERRMSG
UCB\$_ERRCNT

J 13
16-Sep-1984 00:00:33
5-Sep-1984 13:48:56

VAX-11 FORTRAN V3.4-56 Page
DISK\$VMSMASTER:[ERF.SRC]BSDRIVER.FOR;1

6

0001
0002
0003
0004

```
C**Re-written routine, delete old one after testing.
C
```

B

```

0005 Subroutine UCB$B_BS_ERRMSG (lun,ucb$b_bs_errmsg)
0006
0007 byte lun
0008
0009 integer*4 ucb$b_bs_errmsg
0010
0011 Character*(*) Swi_bus, manual, prog, conn, dis_conn, fail
0012 Character*(*) Msg1,msg2,msg3
0013 Character*80 Message
0014
0015 Parameter (
0016 1 Swi_bus = 'SWITCHED BUS, ',
0017 2 Manual = 'MANUAL',
0018 3 Prog = 'PROGRAMABLE ',
0019 4 Fail = 'POWER-FAILURE ',
0020 5 Conn = 'CONNECT TO THIS PORT',
0021 6 Dis_conn = 'DISCONNECT FROM THIS PORT',
0022
0023 1 Msg1 = '"UBA" INITIALIZE IN PROGRESS',
0024 2 Msg2 = 'PORT HAS RECEIVED UNRECOGNIZED INTERRUPT',
0025 3 Msg3 = 'PORT HAS ENCOUNTERED ILLEGAL CONDITION')
0026
0027 call linchk (lun,2)
0028
0029 Goto (10,20,30,40,50,60,70,80) ucb$b_bs_errmsg
0030
0031 write(lun,15) ucb$b_bs_errmsg
0032 15 format(/' ',t8,'UCB$B_BS_ERRMSG',t24,z8.8)
0033 return
0034
0035 10 Message = swi_bus // manual // conn
0036 Length = len (swi_bus) + len (manual) + len (conn)
0037 Goto 999
0038
0039 20 Message = swi_bus // manual // dis_conn
0040 Length = len (swi_bus) + len (manual) + len (dis_conn)
0041 Goto 999
0042
0043 30 Message = swi_bus // fail // dis_conn
0044 Length = len (swi_bus) + len (fail) + len (dis_conn)
0045 Goto 999
0046
0047 40 Message = swi_bus // prog // dis_conn
0048 Length = len (swi_bus) + len (prog) + len (dis_conn)
0049 Goto 999
0050
0051 50 Message = swi_bus // prog // conn
0052 Length = len (swi_bus) + len (prog) + len (conn)
0053 Goto 999
0054
0055 60 Message = msg1
0056 Length = len (msg1)
0057 Goto 999
0058
0059 70 Message = msg2
0060 Length = len (msg2)
0061 Goto 999

```

```
0062
0063      80      Message = msg3
0064      Length = len (msg3)
0065
0066      999      write(lun,998) Message
0067      998      format(/' ',t8,a<length>)
0068
0069      Return
0070      End
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	291	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	387	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	104	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
Total Space Allocated	782	

ENTRY POINTS

Address	Type	Name
0-00000000		UCB\$\$B_BS_ERRMSG

VARIABLES

Address	Type	Name	Address	Type	Name
2-00000050	1*4	LENGTH	AP-00000004	L*1	LUN
2-00000000	CHAR	MESSAGE	AP-00000008	I*4	UCB\$B_BS_ERRMSG

LABELS

Address	Label	Address	Label	Address	Label	Address	Label	Address	Label
0-0000004E	10	1-00000004	15'	0-00000066	20	0-0000007D	30	0-00000094	40
0-000000C2	60	0-000000D4	70	0-000000E6	80	1-00000021	998'	0-000000F6	999
								0-000000AB	50

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name
	LINCHK

```
0001      Subroutine UCB$B_BS_ERRMSG (lun,ucb$b_bs_errmsg)
0002      C
0003      C
0004      C
0005      C      byte          lun
0006      C
0007      C      integer*4      ucb$b_bs_errmsg
0008      C
0009      C
0010      C      call linchk (lun,2)
0011      C
0012      C      if (ucb$b_bs_errmsg .eq. 1) then
0013      C
0014      C10      write(lun,10) 'SWITCHED BUS, MANUAL CONNECT TO THIS PORT'
0015      C      format(/' ',t8,a)
0016      C
0017      C      else if (ucb$b_bs_errmsg .eq. 2) then
0018      C
0019      C      write(lun,10) 'SWITCHED BUS, MANUAL DISCONNECT FROM THIS PORT'
0020      C
0021      C      else if (ucb$b_bs_errmsg .eq. 3) then
0022      C
0023      C      write(lun,10) 'SWITCHED BUS, POWER-FAILURE DISCONNECT FROM THIS PORT'
0024      C
0025      C      else if (ucb$b_bs_errmsg .eq. 4) then
0026      C
0027      C      write(lun,10) 'SWITCHED BUS, PROGRAMABLE DISCONNECT FROM THIS PORT'
0028      C
0029      C      else if (ucb$b_bs_errmsg .eq. 5) then
0030      C
0031      C      write(lun,10) 'SWITCHED BUS, PROGRAMABLE CONNECT TO THIS PORT'
0032      C
0033      C      else if (ucb$b_bs_errmsg .eq. 6) then
0034      C
0035      C      write(lun,10) ''UBA'' INITIALIZE IN PROGRESS'
0036      C
0037      C      else if (ucb$b_bs_errmsg .eq. 7) then
0038      C
0039      C      write(lun,10) 'PORT HAS RECEIVED UNRECOGNIZED INTERRUPT'
0040      C
0041      C      else if (ucb$b_bs_errmsg .eq. 8) then
0042      C
0043      C      write(lun,10) 'PORT HAS ENCOUNTERED ILLEGAL CONDITION'
0044      C      else
0045      C
0046      C15      write(lun,15) ucb$b_bs_errmsg
0047      C      format(/' ',t8,'UCB$B_BS_ERRMSG',t24,z8.8)
0048      C      endif
0049      C
0050      C      return
0051      C
0052      C      end
```

N 13
16-Sep-1984 00:00:33
5-Sep-1984 13:48:56

VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]BSDRIVER.FOR;1 Page 10

COMMAND QUALIFIERS

FORTTRAN /LIS=LISS:BSDRIVER/OBJ=OBJ\$:BSDRIVER MSRC\$:BSDRIVER

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)

/DEBUG=(NOSYMBOLS,TRACEBACK)

/STANDARD=(NOSYNTAX,NOSOURCE FORM)

/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)

/F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOCROSS_REFERENCE /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time:	4.39 seconds
Elapsed Time:	17.23 seconds
Page Faults:	179
Dynamic Memory:	182 pages

